



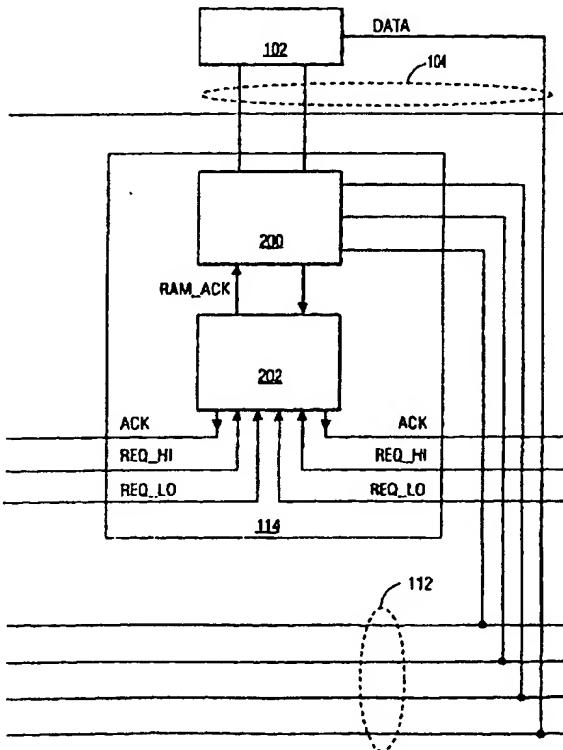
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :	A1	(11) International Publication Number: WO 98/12645
G06F 13/36		(43) International Publication Date: 26 March 1998 (26.03.98)
(21) International Application Number:	PCT/IB97/00876	(81) Designated States: JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date:	15 July 1997 (15.07.97)	
(30) Priority Data:	19 September 1996 (19.09.96) US 08/715,946	
(71) Applicant:	PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).	
(71) Applicant (for SE only):	PHILIPS NORDEN AB [SE/SE]; Kottbygatan 7, Kista, S-164 85 Stockholm (SE).	
(72) Inventors:	JACOBS, Eino; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). TZENG, Tzunghen; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).	
(74) Agent:	DE HAAS, Laurens, J.; Internationaal Octroobureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).	

(54) Title: METHOD AND APPARATUS FOR BUS ARBITRATION WITH WEIGHTED BANDWIDTH ALLOCATION

(57) Abstract

A method and apparatus for bus arbitration with weighted bandwidth allocation are described. Each bus agent is assigned a weight that governs the percentage of bus bandwidth allocated to the agent. An agent is granted control of the bus base, at least in part, upon its weight. The weight corresponds to the number of arbitration states assigned to the agent, where each state represents a grant of bus control. If a first agent is assigned a weight W and all agents together are assigned a total weight Z , an arbiter guarantees bus control to the first agent for at least W arbitrations out of Z arbitrations in which the first agent requests bus control. By employing this scheme, the first agent is guaranteed a fraction W/Z of the bus bandwidth. To ensure flexibility of bandwidth allocation, the weight may be programmed using conventional memory-mapped techniques. The arbitration scheme of the present invention can be split into multiple levels of hierarchy, where arbitration at each level is controlled by an independent state machine. When an agent wins arbitration at one level, it is passed to the next higher level where it competes with other agents at that level for bus access. A bus agent may also raise the priority of its request based upon the urgency of the request. If a low priority request is not acknowledged after the expiration of a predetermined waiting period, then the agent raises the request to a high priority request. The waiting period is selected so that the agent will be guaranteed access to the bus within a worst case latency period after asserting a request.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		

Method and apparatus for bus arbitration with weighted bandwidth allocation.

The invention relates to an information processing device according to the preamble of Claim 1. The invention also relates to an arbiter for arbiting among agents connected to a bus and to a method of arbitrating among agents connected to a bus.

The growing popularity of multimedia software has increased the need for 5 computer systems to handle high-bandwidth, real-time transfers of data. Multimedia systems are distinguished from more traditional computing systems by a high degree of real-time interactivity with the user. This interactivity is accomplished through input/output (I/O) devices, some of which must transfer large volumes of data (e.g., video data) in relatively short periods of time. A computer system must manage the competition of these I/O devices 10 and other functional units for shared data resources, while at the same time assure that the real-time data transfer constraints of the I/O devices and other processor components are satisfied.

Data is communicated among various computer components and peripheral devices over computer buses. A bus may be incorporated onto the microprocessor 15 chip in order to connect the CPU, various caches and peripheral interfaces with each other and ultimately to main memory through an on-chip interface. Buses may also be external to the microprocessor chip, connecting various memory and I/O units and/or processors together in a multiprocessor system. For example, processors may utilize memory as a source of data and instructions, and as a destination location for storing results. Processors 20 may also treat I/O devices as resources for communicating with the outside world, and may utilize buses as communication paths between themselves and memory or I/O devices.

When a bus agent (a device connected to the bus, such as a CPU) wishes to communicate with another agent, the first agent sends signals over the bus that cause the second agent to respond. These signals are collectively called the address or identity. The 25 agent that initiates the communication is called the master, and the agent that responds is called the slave. Some agents act only as masters, some only as slaves, and others as either masters or slaves. If the master's addressing of the slave is acknowledged by the slave, then a data transfer path is established.

Only one agent at a time may communicate over the bus. When two

agents attempt to access the bus at the same time, an arbitration mechanism or protocol must decide which agent will be granted access to the bus. Conventional bus arbitration schemes generally implement a fixed, unchanging priority assignment among the agents. Each agent is assigned a unique priority that remains the same after each round of arbitration. Under this scheme, low priority devices may rarely be granted bus control if they must frequently contend with higher priority devices during each arbitration attempt. This unfairness can be resolved by implementing a round-robin arbitration scheme in which an agent that wins arbitration is reassigned to a very low priority after being granted bus access, thus removing that agent from competition with previously lower priority agents for a period of time.

Some computer systems, at least in multiprocessor technology, implement a mixed arbitration scheme in which bus agents are divided into classes, with each class having a different priority. Devices within a class have the same priority and are generally scheduled to access the bus in a round-robin, equal opportunity manner. Devices that require a high bandwidth and low latency (waiting period between request and grant of bus control) must be assigned to an appropriate priority class to guarantee that the devices are allocated a minimum bandwidth and maximum latency. Although this mixed arbitration scheme is relatively sophisticated, assuring the proper allocation of bus bandwidth using this technique is cumbersome and inflexible. A more flexible system that could more easily be customized to the bandwidth requirements of a particular configuration is desired.

20

The information processing device according to the invention is characterized by the characterizing part of Claim 1. Each bus agent is assigned a weight that governs the percentage of bus bandwidth allocated to the agent. An agent is granted control of the bus based, at least in part, upon its weight. The weight corresponds to the number of arbitration states assigned to the agent, where each state represents a grant of bus control. If a first agent is assigned a weight W and all agents together are assigned a total weight Z , an arbiter of the present invention guarantees bus control to the first agent for at least W arbitrations out of Z arbitrations in which the first agent requests bus control. By employing this scheme, the first agent is guaranteed a fraction W/Z of the bus bandwidth. To ensure flexibility of bandwidth allocation, the weight may be programmed using conventional memory-mapped techniques.

The arbitration scheme of the present invention can be split into multiple levels of hierarchy, where arbitration at each level is controlled by an independent state

machine. When an agent wins arbitration at one level, it is passed to the next higher level where it competes with other agents at that level for bus access. For example, if a first agent occupies a corresponding second level, level 2, and wins arbitration at the second level, then the first agent will contend for arbitration at a first level, level 1, above level 2. The first 5 agent and all other level 2 agents are assigned level 2 priorities and weights. To win arbitration at level 2, the first agent must have the highest level 2 priority among the level 2 agents asserting requests. In general, if the first agent occupies a corresponding k th level and is assigned a k th level weight, then the first agent is granted control of the bus based, at least in part, upon W_k . In particular, where all agents together at the k th level are assigned a total 10 weight Z_k , the first agent is guaranteed bus control for at least W_k arbitrations out of Z_k arbitrations in which the first agent requests bus control and a k th level agent wins bus control. The weight W_k corresponds to W_k arbitration states at the k th level out of a total of Z_k arbitration states at the k th level. This scheme guarantees a fraction W_k/Z_k of the bandwidth at level k to the first agent.

15 If the first agent wins level 2 arbitration, then it is passed on to level 1 as the level 2 winning agent. At level 1, the level 2 winning agent and all other level 1 agents are assigned level 1 priorities and weights. The level 1 priority and weight assigned to the level 2 winning agent are not assigned to the particular level 2 agent that wins an arbitration round, e.g., the first agent, but to the class of level 2 agents that are passed on to level 1. If 20 the level 2 winning agent has a highest level 1 priority among level 1 agents asserting requests, then the level 2 winning agent wins arbitration at level 1 and is granted control of the bus.

The present invention also allows a bus agent to raise the priority of its request based upon the urgency of the request. According to the present invention, a bus 25 agent can indicate the priority of its request to be low or high. When a bus agent wants to initiate a data transfer, it initially posts an adjustable low priority request. If the request is not acknowledged after the expiration of a predetermined waiting period, then the agent raises the request to a high priority request. Generally, the worst case latency period in which the high priority request will be acknowledged is known for a particular computer 30 system. Accordingly, the waiting period is selected so that the agent will be guaranteed access to the bus within the worst case latency period after asserting a request. This priority raising technique of the present invention can be incorporated into any arbitration scheme, and in particular to the weighted arbitration scheme described above.

The objects, features and advantages of the present invention will be apparent to one skilled in the art in light of the detailed description in which the following figures provide examples of the structure and operation of the invention:

Figure 1 illustrates a computer system incorporating the arbitration scheme of the present invention.

Figure 2 is a functional block diagram of the main memory interface of the present invention.

Figure 3 illustrates the major functional blocks of a bus agent for performing the priority raising function of the present invention.

Figure 4 is a state diagram illustrating conventional round-robin arbitration.

Figure 5 illustrates the incorporation of the priority raising function of the present invention into the round-robin arbitration of Figure 4.

Figure 6 is a state diagram illustrating weighted round-robin arbitration according to the present invention.

Figure 7 is a state diagram illustrating another embodiment of weighted round-robin arbitration according to the present invention.

Figure 8 illustrates the incorporation of priority raising into the weighted round-robin arbitration of Figure 6.

Figure 9 illustrates hierarchical arbitration according to the present invention.

Figure 10 is a more detailed illustration of hierarchical arbitration according to the present invention.

The present invention provides a bus arbitration scheme that flexibly allocates bus bandwidth to bus agents. In the following description, numerous details are set forth in order to enable a thorough understanding of the present invention. However, it will be understood by those of ordinary skill in the art that these specific details are not required in order to practice the invention. Further, well-known elements, devices, process steps and the like are not set forth in detail in order to avoid obscuring the present invention.

Figure 1 illustrates the major functional blocks of one embodiment of a

computer system incorporating the arbitration scheme of the present invention. A microprocessor chip 100 is coupled to a main memory device 102 over a main memory bus 104. The main memory 102 may be implemented as a synchronous DRAM (SDRAM). The microprocessor chip 100 includes a central processing unit (CPU) 106 that incorporates an instruction cache 108 and a data cache 110. The CPU 106 and its respective caches communicate with other on-chip components over an internal CPU bus 112. A main memory interface 114 controls the arbitration of various on-chip functional units for control of the internal bus 112, and coordinates the transfer of data between the internal bus 112 and the main memory 102.

A number of the on-chip units provide I/O interfaces employed in multimedia processing. A video input unit 116 receives off-chip video data that can be transferred for storage into the main memory 102 through the bus 112 and the main memory interface 114. A video output unit 118 is responsible for the transfer of video data out of the chip 100 to external I/O units, such as a video display (not shown). Similarly, an audio input unit 120 handles the transfer of audio data into the chip 100, whereas an audio output unit 122 coordinates the transfer of audio data from the chip 100 to an off-chip audio unit, such as a sound card (not shown).

The microprocessor further includes an image co-processor 124, which is dedicated to performing complex image processing tasks that would otherwise occupy the CPU 106 for long periods of time. A VLD (Variable Length Decoder) co-processor 126 is used to speed up computation of the MPEG algorithm preferably employed to decompress video data. Further, a PCI (Peripheral Component Interconnect) interface unit 128 permits the on-chip units to be coupled to a PCI bus. Finally, boot unit 130 loads main memory 102 with a boot routine from an external EPROM upon power-up or reset.

Figure 2 illustrates a functional block diagram of the main memory interface 114. The main memory interface 114 includes a memory controller 200 and an arbiter 202. The arbiter 202 determines which bus agent that contends for access to the internal CPU bus 112 will be granted control of the bus 112. The memory controller 200 coordinates the transfer of data between that agent and other bus agents or the main memory 102.

General Protocol

The general protocol employed by the present invention to perform a main memory transfer over the internal bus 112 may be described, in one embodiment, as follows:

1. A bus master asserts a request for control of the bus 112. As described below, the present invention employs two request signals: a high priority request REQ_HI and a low priority request REQ_LO. The memory controller 200 issues a START signal to indicate that it is ready to initiate a transfer, which requires the arbiter to perform an arbitration.

5

2. In the same cycle or later, the arbiter 202 responds to the bus master by asserting an acknowledgment signal ACK. This signal indicates that the internal bus 112 is available to the requester and that the request will be handled. If the bus is occupied, the acknowledgment will be delayed. Similarly, the arbiter 202 asserts a RAM_ACK signal to

10 the memory controller 200 after a request has been received and successfully arbitrated.

3. The requester responds to the ACK signal by transmitting an address over a tri-state address bus that is shared with all other bus agents. The address indicates the main memory address associated with the transfer. Simultaneously, the requester indicates the type of transfer (read or write) using a tri-state opcode bus that is also shared with all other bus agents. The arbiter 202 deasserts ACK in this cycle.

15

4. After deassertion of ACK, the requester deasserts the request signal, while the address and opcode signals remain asserted until a transfer signal is asserted.

5. After a main memory latency period, the memory controller 200 asserts the transfer signal. The transfer signal may come one cycle after the ACK signal or it may 20 come later.

6. One cycle after transfer, the first word of a block of data is transferred over the data bus between the bus agent and the main memory 102. In this cycle, all control signals are deasserted, and the address and opcode buses are tri-stated.

7. In subsequent cycles a sequence of word transfers occurs to complete the 25 rest of the block transfer between the bus agent and the main memory 102. The block size is constant and hard-coded in the design of the memory controller 200 and the bus agents. The transfer order is provided by the signal opcode (read or write). Accordingly, both the bus agent and the memory controller 200 are informed of the block size and the transfer order, so no further handshaking is necessary to complete the bus transaction.

30 The protocol for coordinating memory-mapped I/O transfers is essentially the same as that for main memory transfers. An example of a memory-mapped I/O transfer is a transfer between the data cache 110 and a control register in the video input unit 116. For memory-mapped I/O, the memory controller 200 asserts an MMIO signal (not shown) after ACK to indicate to all devices on the bus 112 that an MMIO transaction is starting.

After MMIO is asserted, every MMIO device inspects the address on the bus 112 to determine whether it is being addressed. The addressed device asserts an MMIO REPLY signal (not shown) to the arbiter to indicate that it is ready to complete the MMIO transfer.

5 Priority Raising

With this background in place, the priority raising function of the present invention will now be described. Generally, the best CPU performance is obtained if cache misses take priority over I/O traffic on the internal bus 112. However, cache priority must be balanced against the competing real-time constraints of the I/O units. For example, a 10 video output device must be granted control of the bus within a maximum, worst case latency period in order to provide a high quality image to an external display.

Figure 3 illustrates the major functional blocks of a bus master 300 for performing the priority raising function of the present invention. The relevant blocks in the bus master 300 include a time-out register 302, a timer circuit 304 and a control logic circuit 15 306. The time-out register 302 stores a time-out value. The time-out register 302 can store a fixed time-out value or be programmed according to conventional memory-mapped techniques.

An I/O device or other unit in the computer system of the present invention can indicate the priority of its requests to be low or high. Cache requests and 20 urgent I/O requests, such as from the image co-processor, should be assigned a high priority. Less urgent I/O requests should be assigned a low priority. When a low priority bus agent 300 wants to initiate a data transfer, the control unit 306 initially posts an adjustable low priority request REQ_LO. The control unit 306 simultaneously issues a start signal to the timer 304 to start a countdown of the timer 304. The time-out or waiting period stored in the 25 time-out register is chosen so that the agent 300 will be guaranteed access to the bus within the worst case latency period after asserting a request. The time-out period is typically expressed in processor clock cycles, and is selected as the worst case latency period less the worst case waiting time for a high priority request to win arbitration.

If no acknowledgment from the arbiter 202 has been received within the 30 time-out period, then the timer 304 issues a time-out signal to the control unit 306. In response, the control unit 306 raises the request to a high priority request REQ_HI. Generally, in an arbitration scheme such as round-robin, agent 300 will then win arbitration over other high priority devices. The other devices typically will have been granted bus access more recently than agent 300, thereby causing them to be rotated to lower priorities

than agent 300 according to the round-robin algorithm. Further, a high priority request from agent 300 will, of course, win arbitration over a low priority request. Priority raising therefore guarantees bus access to agent 300 within the worst case latency period.

Priority raising can be incorporated into any arbitration scheme. For example, Figures 4 and 5 illustrate priority raising in round-robin arbitration. Figure 4 diagrams conventional round-robin arbitration. In state A, bus agent A has control of the bus, whereas in state B, bus agent B has control. The arc from state A to state B indicates that when agent A owns the bus, and a request from agent B is asserted, then a transition to state B occurs, i.e., ownership of the bus passes from agent A to agent B. When the arbitration is in state A and agent A asserts a request while agent B does not, then agent A retains control of the bus. When the arbitration is in state A and both agents A and B assert requests, then ownership of the bus transfers to agent B, creating fair allocation of ownership.

Arbitration state transitions for the round-robin scheme or any other scheme can be viewed in terms of priorities. Referring to Figure 4, when in state A, agent B has a higher round-robin priority than agent A, i.e., if both A and B assert requests, then ownership passes to B. After the transition, the agent (B) granted control is rotated to the lowest round-robin priority in the priority order. As a result, A now is assigned the highest round-robin priority, and A will gain control of the bus if both A and B assert requests. In this manner, the round-robin scheme can be viewed as rotating the round-robin priority order after each arbitration.

Figure 5 illustrates the incorporation of priority raising into the simple round-robin example of Figure 4. Assume that bus agent A is assigned a fixed high priority. For example, bus agent A may be an instruction cache or a data cache, which should have a minimum latency in order to achieve optimum CPU performance. Further, assume that bus agent B is an I/O device that incorporates priority raising circuitry, as shown in Figure 3.

Referring to Figure 5, if A has control of the bus and B asserts a low priority request while A does not assert a request, then B wins the arbitration and is granted control of the bus. However, if A has control and B asserts a low priority request while A asserts its high priority request, then A is again granted control of the bus. This situation may continue for many arbitration cycles, essentially shutting out B from access to the bus. According to the priority raising mechanism, after a predetermined waiting period, B will raise its request to a high priority request. At that time A and B will compete equally in the round-robin scheme, and control will pass to B even if A is simultaneously asserting a high

priority request.

Based on this example, it can be seen that, in general, agent A wins arbitration if it asserts a high priority request while agent B asserts a low priority request. If both A and B assert requests of the same priority, then arbitration is resolved in the conventional manner. Looked at another way, agent B wins arbitration if both agents A and B assert high priority requests and agent B would have won arbitration if both A and B were asserting low priority requests.

Weighted Round-Robin Arbitration

Priority raising is but one technique employed in the arbitration scheme of the present invention. In addition, or as an independent alternative, the present invention modifies the conventional round-robin scheme to account for the fact that the bandwidth and latency requirements of the bus agents differ. As discussed above, the caches should be allocated the greatest share of bus bandwidth, and thus the minimum latency, because the best CPU performance is obtained if cache misses are given the highest priority access to the bus. In contrast, an audio device operates at a relatively low bandwidth and can wait a relatively long time for a data transfer.

According to another embodiment of the present invention, the bus agent priorities are weighted so that the agents may be allocated unequal shares of bandwidth during round-robin arbitration. Figure 6 is a state diagram illustrating weighted round-robin arbitration in which bus agent A is allocated twice as much bandwidth as bus agent B. According to the usual round-robin scheme, bus agent A would be reassigned to a low (preferably the lowest) round-robin priority after winning a first round of arbitration. However, in the example of Figure 6, bus agent A is assigned a weight of 2. This double weight indicates that bus agent A can retain its high priority status for a total of two arbitration rounds out of the three rounds represented by the three state transition nodes A1, A2 and B. Accordingly, after bus agent A wins the first round of arbitration (state A1), then bus agent A would win a second round of arbitration if A again requests access to the bus (state A2). If, however, during this second round, A does not request bus access but B does, then bus agent B would win the second round of arbitration. Because A is only assigned a weight of 2, then after state A2 (in which A has won arbitration for two rounds), B would win the next arbitration round if B requests bus access. In general, if the total weight assigned to all bus agents is Z, then a bus agent having a weight W will be assigned the highest priority for at least W arbitration rounds out of Z arbitration rounds in which the

agent requests bus access.

Figure 7 is a state diagram illustrating a more complicated implementation of the weighted round-robin arbitration scheme of the present invention. The bus agents A, B and C are proportionately weighted according to the ratio 2:1:1. Assuming that all agents are requesting bus access, the state transition sequence is A1, B, A2, C. Here, the total weight Z=4. Because of this weighting, agent A can retain the highest priority for at least two out of four arbitration rounds in which A requests bus control.

Weighted round-robin arbitration can be combined with the priority raising feature of the present invention. Figure 8 illustrates priority raising incorporated into the weighted round-robin arbitration of Figure 6. In the case of Figure 6, where the agents can assert only a single-level priority, if both A and B assert requests starting at state A1, then A wins the arbitration through a transition to state A2. However, according to Figure 8, if one of the agents asserts a high priority request (after raising it from an adjustable low priority) and the other agent asserts either no request or a low priority request, then the high priority requesting agent wins the arbitration round. For example, starting at state A1, B wins the arbitration if B raises its adjustable low priority request to a high priority request (BH) and A asserts either no request or a low priority request (AL). Similarly, at state A2 if A issues a high priority request (AH) and B issues either no request or a low priority request (BL), then A remains at state A2, even though under the round-robin scheme of Figure 6 arbitration would have transitioned to state B. In the case where both A and B assert requests of the same priority level, arbitration follows the state transition diagram of Figure 6. Further, an agent asserting even a low priority request, of course, wins arbitration if no other agent asserts any request at all.

25 Arbitration Hierarchy

The arbitration scheme of the present invention can be split into multiple levels of hierarchy, as shown in Figure 9. Each level of hierarchy constitutes an independent arbitration state machine, as generally illustrated in Figure 10. When a device wins arbitration at one level, it is passed to the next level where it competes with other devices at 30 that level for bus access. This process is continued until the highest level of arbitration, where an agent ultimately wins control of the bus.

Figure 9 illustrates an example of a weighted round-robin, four-level arbitration hierarchy according to the present invention. Each device of Figure 1 is assigned to a hierarchical level and weighted within its assigned level. Memory-mapped I/O (MMIO),

data cache and instruction cache devices preferably are arbitrated with fixed weights among each other (i.e., 1) under control of a cache arbiter 900. Preferably, each of these devices can only issue a high priority request REQ_HI. At level 1 902, the winner of the cache arbitration is assigned a programmable weight of 1, 2 or 3. The winner of the cache arbitration contends for the bus at level 1 900 with the winner of level 2 arbitration, the level 2 winner having a programmable weight of 1, 2 or 3 at level 1 902. The requests surviving the level 2 arbitration can have a low or high priority.

Level 2 904 contains the image co-processor (ICP) 124 and the PCI bus interface 128. The image co-processor 124 preferably is assigned a programmable weight of 1, 3 or 5, whereas the PCI bus is assigned a weight of 1. These devices contend with the winner of level 3 arbitration. At level 2, the level 3 arbitration winner is preferably assigned a programmable weight of 1, 3 or 5.

Level 3 906 contains high-bandwidth video devices: video-in 116, video-out 118 and the VLD co-processor 126. The YUV video components of the video-in signal contend for arbitration in a round-robin YUV arbiter 908. Similarly, the YUV components of the video-out signal contend for arbitration in a round-robin YUV arbiter 910. The Y video component is preferably assigned a weight of 2 because it carries the most video information, whereas the U and V components are each assigned a weight of 1. Each combined YUV signal has a weight of 2 at level 3 906. The video devices contend at level 3 with the winner of level 4 arbitration, which is assigned a level 3 weight of 1.

Level 4 912 contains low-bandwidth devices, including the audio units 120 and 122 and the boot unit 130. The audio units and the boot unit are each preferably assigned weights of 1.

Figure 10 illustrates a portion of the arbitration hierarchy of Figure 9 in greater detail. The arbitration at each level is implemented in a state machine. If programmable weighting is employed at a particular level, then arbitration at that level should be implemented using a programmable state machine. Programmable state machines are well known in the art, and may be embodied in a programmable logic array (PLA) or a similar device. If fixed weighting is desired, then fixed logic may be utilized also. Arbitration weights are assigned by giving a device a number of state nodes in the arbitration state machine equal to the weight of the device. For programmable weights, nodes in the state machine may be activated or deactivated.

According to the example of Figures 9 and 10, a significant variation in bandwidth that would require programmable weighting is only anticipated for the device

types at the first two levels. Adequate performance can be achieved by employing fixed weights for the third and fourth levels. Those skilled in the art will understand that the programmable or unprogrammable nature of the state machines can be varied in design to accommodate different expectations of variation in bandwidth.

5 The weights, and thus the bandwidth, of devices at the first and second levels can be programmed by writing the desired weights into a memory-mapped bandwidth control register 1002. In this example, the bandwidth control register 1002 contains four fields to select the weights for the two respective winners of the cache arbitration and the level 2 arbitration at level 1 902, the weight of the image co-processor at level 2 904, and
10 the weight at level 2 904 of the winner of the level 3 906 arbitration. As mentioned above, changing the weight of a device activates or deactivates nodes in the state machine. For example, the weight of agent A in Figure 6 would be changed from 2 to 1 by deactivating node A2, which would result in the state diagram of Figure 4.

Figure 10 also illustrates that the request lines to each state machine are
15 generally divided into high and low priority requests. A device identification number identifying the device winning a lower level arbitration is passed to the next level along with the high or low priority request from that device. Note that not all the request lines shown in Figure 9 are detailed in Figure 10.

In general, each of the state machines of Figure 10 preferably performs
20 weighted round-robin arbitration with priority raising. When an agent wins arbitration at one level, it is passed on to the next higher level to contend for arbitration at that level. For example, the image co-processor 124 contends for arbitration at level 2 904 with PCI interface 128 and the winner of the level 3 arbitration. The level 2 state machine 904 must consider a number of factors to determine whether the image co-processor 124 wins level 2
25 arbitration: the round-robin priority at level 2 of the image co-processor 124 compared to the level 2 round-robin priority of other level 2 agents issuing requests; and whether the image co-processor 124 is asserting an adjustable low or high priority request according to the priority raising technique of the present invention. If, after considering these factors, the level 2 state machine 904 determines that the image co-processor 124 wins arbitration at
30 level 2, then the image co-processor 124 request is presented to the level 1 state machine 902 as the request of the level 2 winning agent.

At level 1 902, the level 2 winning agent contends for arbitration with the winner of the cache arbitration. To determine whether the level 2 winning agent wins arbitration at level 1, the level 1 state machine 902 must consider the following factors: the

round-robin priority of the level 2 winning agent at level 1 compared to the level 1 priority of the winner of the cache arbitration; and whether the level 2 winning agent is asserting an adjustable low priority or high priority request according to priority raising. The winner of the level 1 arbitration will be granted control of the bus. It is important to note the
5 distinction between winning arbitration at a particular level and ultimately being granted control of the bus, which only occurs upon winning level 1 arbitration.

In this example, if the image co-processor 124 is granted control of the bus, then at the "home" level 2 904 the level 2 state machine 904 will experience a transition to the next state during the next round of arbitration. At level 2 904, the image co-processor
10 124 occupies W_2 state transition nodes out of Z_2 nodes, where W_2 is the level 2 weight of the image co-processor 124 and Z_2 is the total level 2 weight of all the devices at level 2. Assuming no priority raising for the sake of this example, this configuration guarantees bus control to the image co-processor 124 for at least W_2 arbitrations out of Z_2 arbitrations in which the image co-processor 124 requests bus control and a level 2 agent wins bus control.

15 At level 1, the granting of bus control to the level 2 winning agent also causes the level 1 state machine 902 to experience a transition to the next state. The level 2 winning agent occupies W_1 state transition nodes out of Z_1 nodes at level 1, where W_1 is the level 1 weight of the level 2 winning agent and Z_1 is the total level 1 weight of all devices at level 1. This configuration guarantees bus control to the level 2 winning agent for
20 at least W_1 arbitration rounds out of Z_1 rounds in which the level 2 winning agent requests bus control.

It is important to note that the level 2 winning agent refers to the class of level 2 agents at level 1 that win level 2 arbitration, and not to the individual level 2 agent that happens to win a particular arbitration round. It is the level 2 input to the level 1 902
25 state machine that experiences a transition in the level 1 902 state machine, and not just the particular level 2 agent that happens to win an arbitration round, e.g., the image co-processor 124.

Bandwidth Allocation

30 Bandwidth is allocated at every level relative to the weights of the devices. The fraction of bandwidth of a device x is:

$$F_x = W_x/Z_L,$$

where W_x is the weight of device x , and Z_L is the sum of the weights of all devices at the level L where the device x resides. For example, level 4 occupies 1/6th of the bandwidth of

level 3.

The guaranteed minimum bandwidth for device x is:

$$B_x = F_x \times B_L,$$

where B_L is the total bandwidth available at level L.

5 The expected available bandwidth for a device differs from the guaranteed minimum bandwidth, depending on the application. If a particular device does not use all of its bandwidth, then other devices at the same level will get correspondingly more bandwidth. If bandwidth is not all used at a level, then higher levels will be able to employ more bandwidth.

10 Minimum bandwidth is closely related to maximum latency. The maximum latency L_x for device x is:

$$L_x = \text{ceil} (Z_L/W_x) \times (B_{\text{tot}}/B_L - 1) \times T \text{ (clock cycles)},$$

where B_{tot} is the total bus bandwidth,

ceil is the ceiling or next highest integer function, and

15 T is the transfer time of one transaction ($T = 16$ cycles if main memory bandwidth is four bytes per cycle and the transfer size is 64 bytes).

Note that expected latency is normally much lower than the worst case maximum latency because rarely do many devices issue requests at exactly the same time.

Given the number of factors involved, the programming of the arbitration 20 weights is best performed by first assuming different sets of weights and determining the resultant bandwidths for the corresponding devices. Then, the optimum set of weights is selected based upon the corresponding resultant bandwidths that most closely match the desired bandwidth allocation.

For example, assume a computer system having 400 MB/s main memory 25 bandwidth and a transfer time of $T = 16$ cycles. Further assume a 1:1 bandwidth weighting at level 1, and a 1:1:1 bandwidth weighting at level 2. The remainder of the bandwidth weighting follows the fixed weighting scheme of Figure 9. This weighting results in the following bandwidth allocation to the different levels of hierarchy:

	Level 1:	200MB/s
30	Level 2:	133MB/s
	Level 3:	56MB/s
	Level 4:	11MB/s

For some individual devices, bandwidth and latency are as follows:

MMIO

(Assume no instruction or data cache misses)

Bandwidth = $1/2 \times 400 = 200\text{MB/s}$

Maximum latency = $(2/1 - 1) \times 16 = 16 \text{ cycles}$

Instruction cache, data cache

5

(Assume only one cache miss, no MMIO accesses)

Bandwidth = $1/2 \times 400 = 200\text{MB/s}$

Maximum latency = $(2/1 - 1) \times 16 = 16 \text{ cycles}$

Image Co-processor

10

(Assume all units issue requests at maximum rate)

Bandwidth = $1/3 \times 200 = 66\text{MB/s}$

Maximum latency = $(3/1 \times 400/200 - 1) \times 16 = 80 \text{ cycles}$

VLD

15

(Assume all units issue requests at maximum rate)

Bandwidth = $1/6 \times 1/3 \times 200 = 11\text{MB/s}$

Maximum latency = $(6 \times 400/67 - 1) \times 16 = 560 \text{ cycles}$

Audio

20

(Assume all units issue requests at maximum rate)

Bandwidth = $1/3 \times 1/6 \times 1/3 \times 200 = 3.7\text{MB/s}$

Maximum latency = $(3/1 \times 36 - 1) \times 16 = 1,712 \text{ cycles}$

As an example, Table 1 illustrates percentage bandwidth allocation among caches and peripheral units at level 1. Table 2 illustrates bandwidth allocation among the image co-processor, the PCI interface and the winner of the level 3 arbitration.

Table 1. Bandwidth allocation among caches and peripheral units.

	weight of MMIO and caches	weight of level 2	bandwidth at level 1	bandwidth at level 2
5	3	1	75%	25%
	2	1	67%	33%
	3	2	60%	40%
10	1	1	50%	50%
	2	3	40%	60%
	1	2	33%	67%
	1	3	25%	75%

15 Table 2. Bandwidth allocation among ICP, PCI and devices at level 3.

	weight of ICP	weight of level 3	bandwidth for ICP	bandwidth at level 3	bandwidth for PCI
20	1	1	33%	33%	33%
	3	1	60%	20%	20%
	5	1	72%	14%	14%
25	1	3	20%	60%	20%
	3	3	43%	43%	14%
	5	3	56%	33%	11%
	1	5	14%	72%	14%
30	3	5	33%	56%	11%
	5	5	45%	45%	10%

Although the invention has been described in conjunction with a number of embodiments, those skilled in the art will appreciate that various modifications and alterations may be made without departing from the spirit and scope of the invention. For example, although for purposes of explanation the following description provides examples of arbitration for an internal CPU bus, it will be understood by those of ordinary skill in the art

that the present invention is generally applicable to the control of any communications bus, as well as to the accessing of any common resource. Further, those skilled in the art will understand the principles disclosed herein are applicable to systems having any number of bus agents, any number of weights per bus agent, any number of hierarchical levels and any 5 number of priority levels for each request.

CLAIMS:

1. An information processing device comprising
 - a bus;
 - agents coupled to the bus;
 - an arbiter for deciding which of the agents will be granted control of the bus in case more than one of the agents requests control of the bus at the same time, characterized in that the arbiter decides which of the agents will be granted control based upon relative weights assigned to respective ones of the agents.
2. An information processing device as claimed in Claim 1, wherein a first agent among said agents is assigned a weight W and all agents together are assigned a total weight Z, the arbiter guaranteeing bus control to the first agent for at least W arbitrations out of Z arbitrations in which the first agent requests control of the bus.
3. An information processing device as Claimed in Claim 1, wherein a first agent among said agents is assigned a weight W and all agents together are assigned a total weight Z, the arbiter having has at least Z arbitration states, each arbitration state representing a grant of bus control to a corresponding one of the agents.
4. An information processing device as Claimed in Claim 3, the arbiter giving priority to transitions to the arbitration states according to a round robin scheme.
5. An information processing device as Claimed in Claim 1, the arbiter grouping the agents in levels, an agent winning arbitration among the agents at a kth level contending for arbitration at a higher k-1th level.
6. An information processing device as Claimed in Claim 1 a first agent among the agents having means to assert a low priority request and a high priority request for control of the bus, the first agent raising the low priority request to the high priority request if the low priority request is not granted after a predetermined waiting period.
7. An arbiter for arbiting between agents connected to a bus, the arbiter deciding which of the agents will be granted control of the bus in case more than one of the agents requests control of the bus at the same time, characterized in that the arbiter decides which of the agents will be granted control based upon relative weights assigned to respective ones of the agents.

8. A method of arbitrating among agents connected to a bus in case more than one of the agents requests control of the bus at the same time, characterized in that it is decided which of the agents will be granted control based upon relative weights assigned to respective ones of the agents.
- 5 9. A method as claimed in Claim 8, wherein a first agent among said agents is assigned a weight W and all agents together are assigned a total weight Z, the method guaranteeing bus control to the first agent for at least W arbitrations out of Z arbitrations in which the first agent requests control of the bus.
10. A method as Claimed in Claim 8, the agents being grouped in levels, an
10 agent winning arbitration among the agents at a kth level contending for arbitration at a higher k-1th level.

1/7

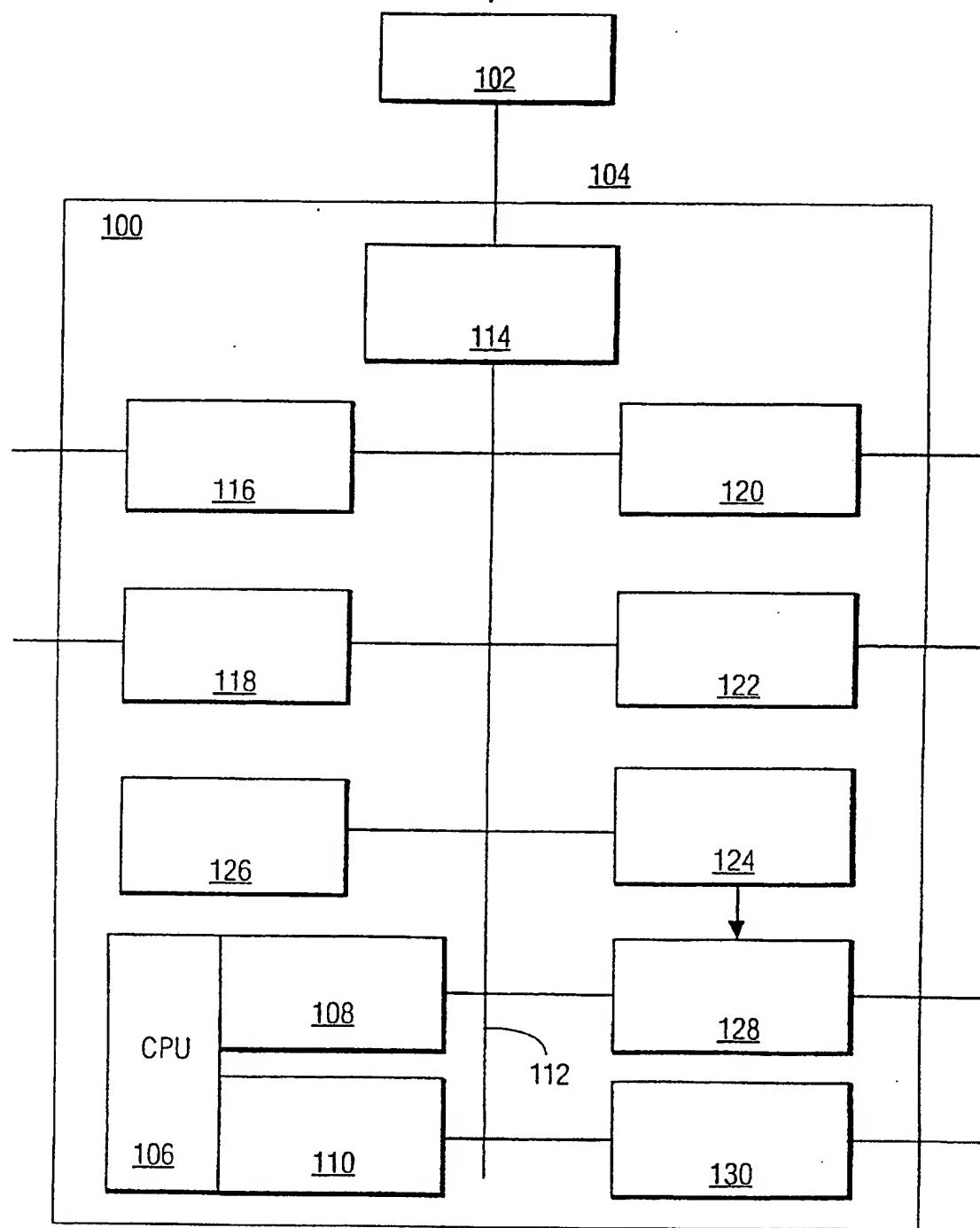


FIG. 1

2/7

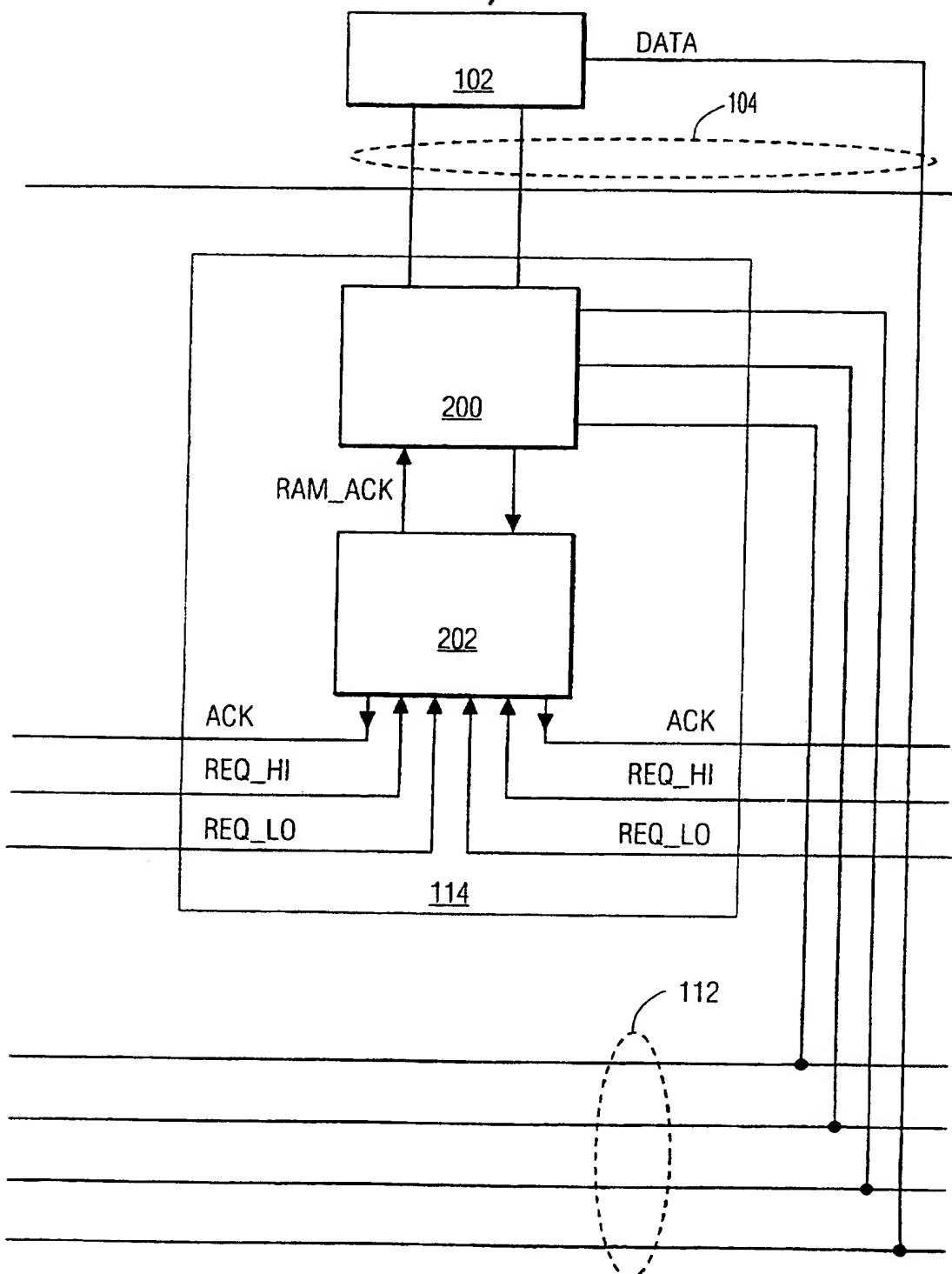


FIG. 2

3/7

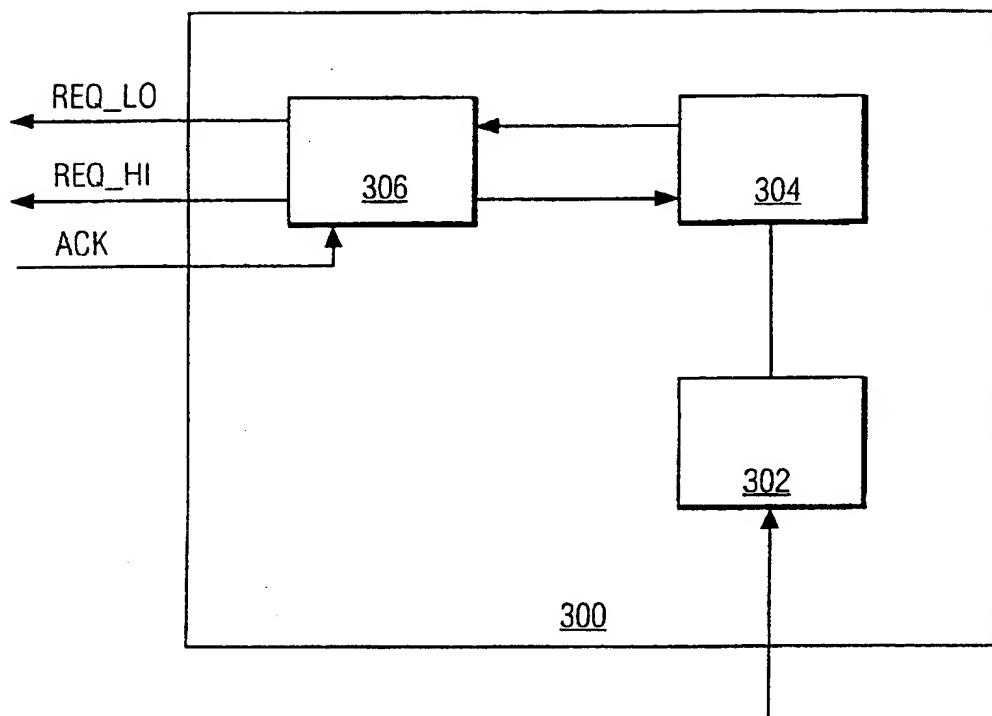


FIG. 3

4/7

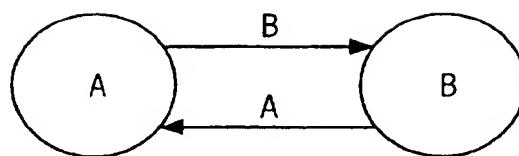


FIG. 4

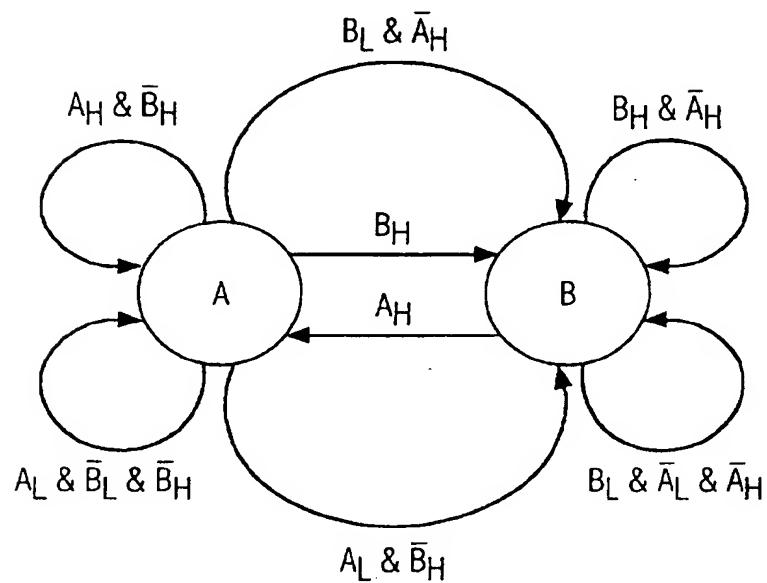


FIG. 5

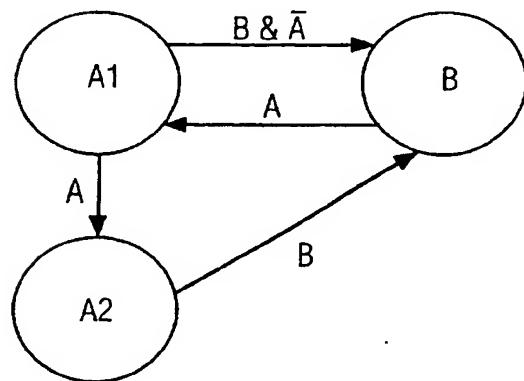


FIG. 6

5/7

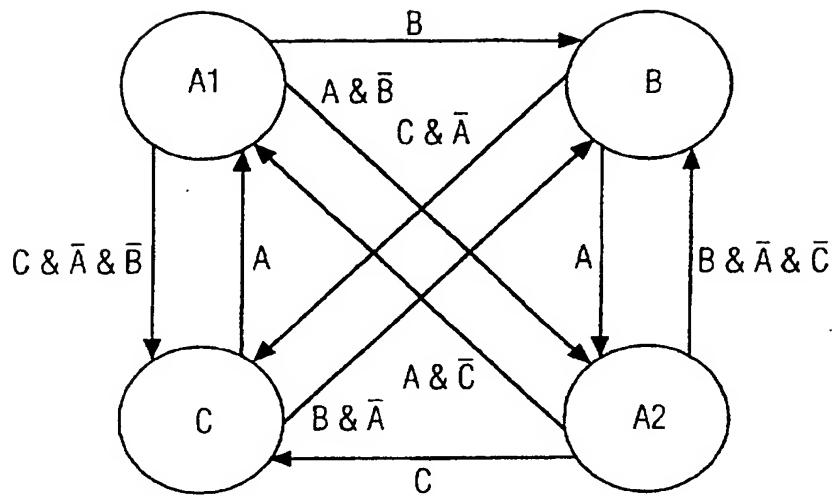


FIG. 7

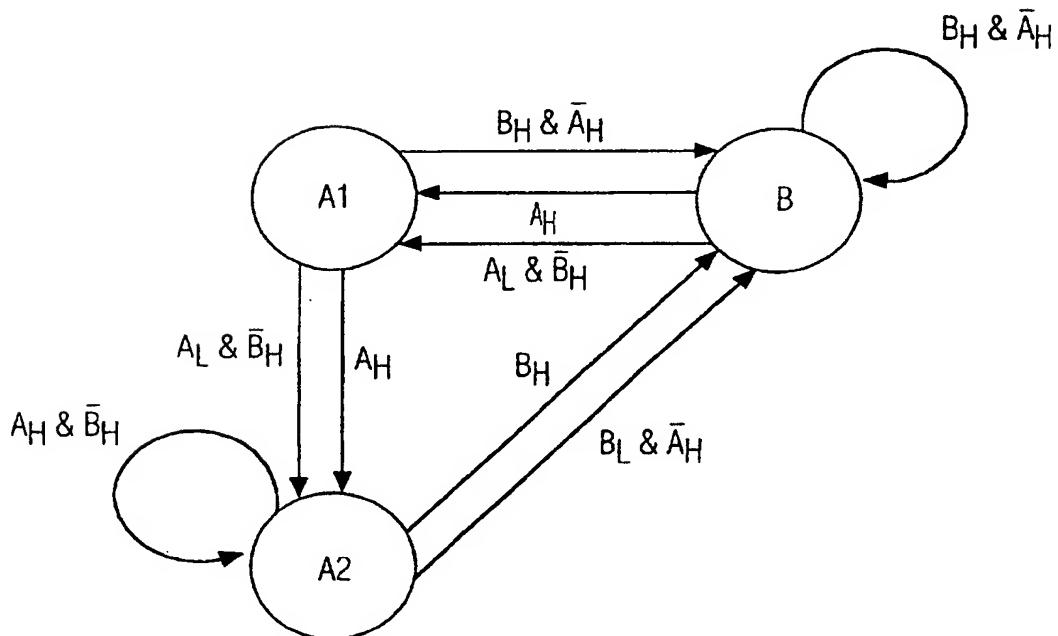


FIG. 8

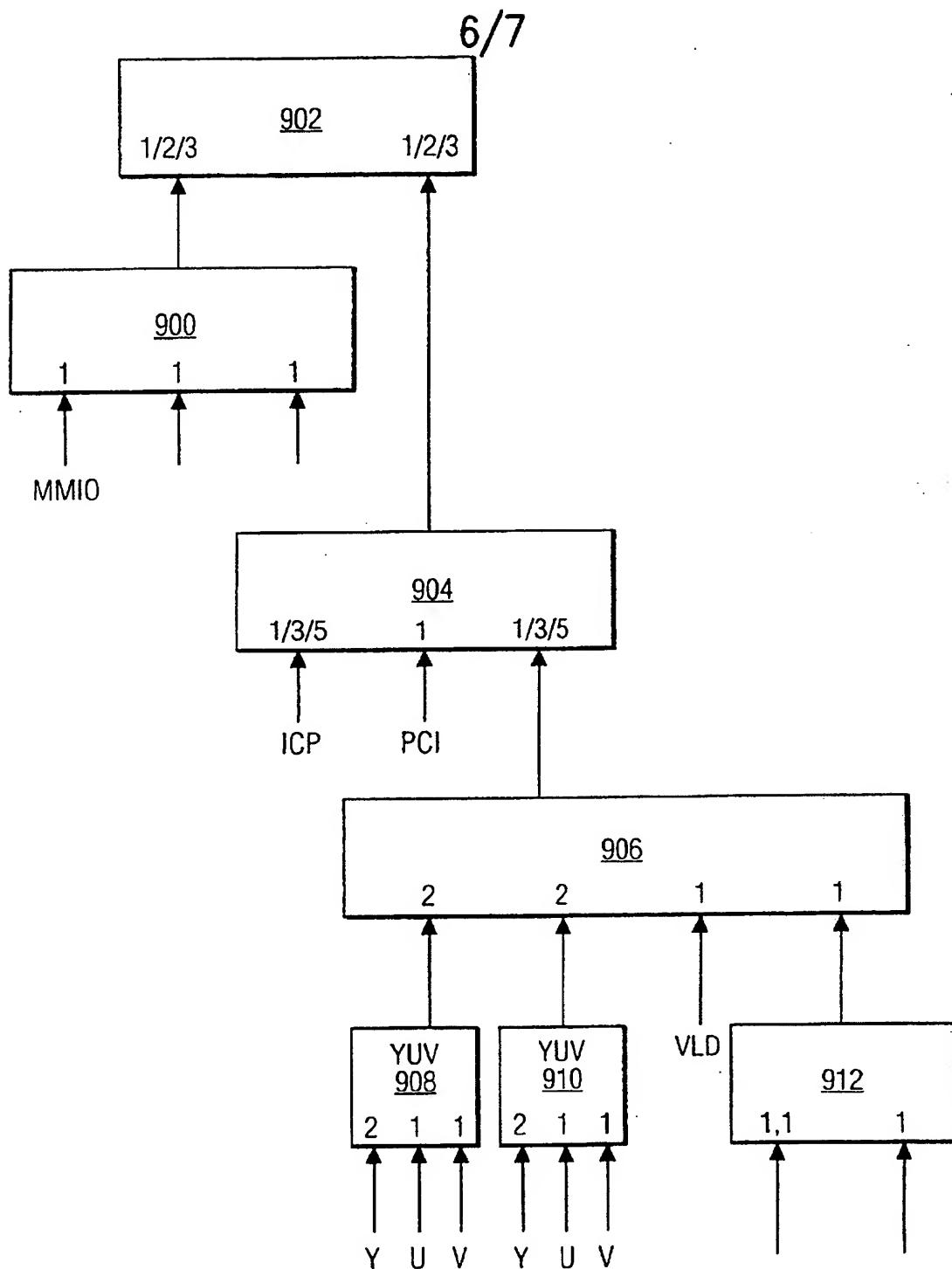


FIG. 9

7/7

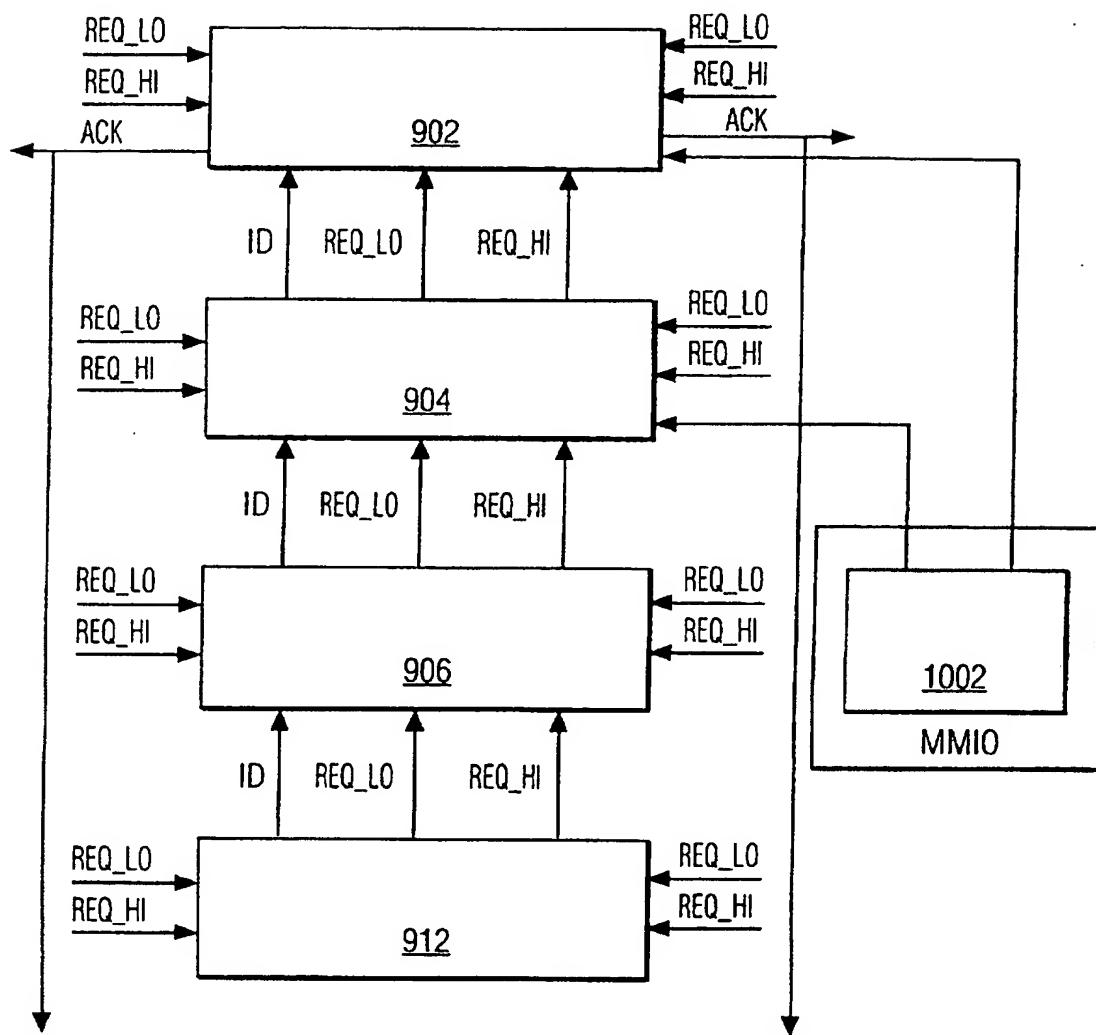


FIG. 10

INTERNATIONAL SEARCH REPORT

1

International application No.

PCT/IB 97/00876

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: G06F 13/36

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 8095906 A (MATSUSHITA ELECTRIC IND CO LTD), 12 April 1996 (12.04.96)	1-4,7-9
Y	--	5,6,10
X	US 5463624 A (RAYMOND L. HOGG ET AL), 31 October 1995 (31.10.95), column 12, line 1 - column 24, line 56, abstract	1,7,8
Y	--	5,6,10
A	US 5526496 A (HUSSEIN M. ALNUWEIRI), 11 June 1996 (11.06.96), abstract	1
	--	

 Further documents are listed in the continuation of Box C. See patent family annex.

- * Special categories of cited documents:
- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed
- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

5 February 1998

Date of mailing of the international search report

06-02-1998

Name and mailing address of the ISA/
Swedish Patent Office
Box 5055, S-102 42 STOCKHOLM
Facsimile No. +46 8 666 02 86

Authorized officer

Björn Edlund
Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB 97/00876

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4621342 A (GIUSEPPE N. CAPIZZI ET AL), 4 November 1986 (04.11.86), abstract ---	1
A	IEEE, PROCEEDINGS, SOUTHEASTCON'90, pp 750-754, Volume 2, April 1990, R.G. Deshmukh, S. Doi, "The Multiprocessor System With A Fairly Weighted Global Bus Arbitration Circuit", page 751 - page 752, Scan Global Bus Arbitration Circuit ---	1

INTERNATIONAL SEARCH REPORT
Information on patent family members

07/01/98

International application No. PCT/IB 97/00876	
---	--

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
JP 8095906 A		12/04/96	NONE		
US 5463624 A		31/10/95	CA 2187291 A	26/10/95	
			EP 0760185 A	05/03/97	
			FI 964119 A	13/12/96	
			IL 113370 D	00/00/00	
			WO 9528786 A	26/10/95	
US 5526496 A		11/06/96	US 5572687 A	05/11/96	
US 4621342 A		04/11/86	NONE		